3-14-2020

Development and Validation of the Computational Thinking Concepts and Skills Test

Markeya S. Peteranetz

Patrick M. Morrow

Leen-Kiat Soh

Development and Validation of the Computational Thinking Concepts and Skills Test

Markeya S. Peteranetz
College of Engineering
University of Nebraska-Lincoln
Lincoln, NE, USA
peteranetz@unl.edu

Patrick M. Morrow
Dept. of Computer Science & Engineering
University of Nebraska-Lincoln
Lincoln, NE, USA
pmorrow4@unl.edu

Leen-Kiat Soh
Dept. of Computer Science & Engineering
University of Nebraska-Lincoln
Lincoln, NE, USA
lksoh@cse.unl.edu

ABSTRACT

Calls for standardized and validated measures of computational thinking have been made repeatedly in recent years. Still, few such tests have been created and even fewer have undergone rigorous psychometric evaluation and been made available to researchers. The purpose of this study is to report our work in developing and validating a test of computational thinking concepts and skills and to compare different scoring methods for the test. This computational thinking exam is intended to be used in computing education research as a common measure of computational thinking so that the research community will be able to make more meaningful comparisons across samples and studies. The Computational Thinking Concepts and Skills Test (CTCAST) was administered to students in several courses, evaluated and revised, and then administered to another group of students. Part of the revision included changing half of the items to a multiple-select format. The test scores using the three scoring methods were compared to each other and to scores on a different test of core computer science knowledge. Results indicate the CTCAST and the test of core computer science knowledge measure similar, but not identical, aspects of students' knowledge and skills, and that item-level statistics vary according to the scoring method that is used. Recommendations for using and scoring the test are presented.

CCS CONCEPTS

• **General and reference~Measurement** • **Social and professional topics~Computational thinking** • **Social and professional topics~Student assessment** • Social and professional topics~Computational science and engineering education

KEYWORDS

Assessment; Computational thinking; Multiple-select items; Test development

SIGCSE '20, March 11–14, 2020, Portland, OR, USA

ACM 978-1-4503-6793-6/20/03…$15.00
https://doi.org/10.1145/3328778.3366813

ACM Reference format:

Markeya S. Peteranetz, Patrick M. Morrow, and Leen-Kiat Soh. 2020. Development and validation of the computational thinking concepts and skills test. In *Proceedings of ACM SIGCSE conference (SIGCSE'20)*, March 11-14, 2020, Portland, OR, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3328778.3366813.

1 INTRODUCTION

Increased interest in recent years in computational thinking [20] (CT) as a set of foundational cognitive processes that are crucial to success in computing disciplines and complementary to broader problem solving skills has led to a corresponding increase in research around CT. Along with the growth of research on CT has come awareness of the need to effectively and efficiently measure CT. In recent years, scholars have called for standardized and validated measures of CT that would allow researchers working in varied contexts to have a common indicator of CT [4, 10, 12]. Furthermore, there is recognition that CT skills are not the same thing as the knowledge and skills acquired through formal computer science (CS) education—there is overlap, but there are also differences. The differences between CT and CS content knowledge make it so that tests that are aligned with CS course content are not necessarily appropriate measures of CT. As a result, there have been multiple efforts to construct tests of CT (described in Section 2.1). There is not yet a consensus as to what exactly constitutes CT and as a result, different test makers have adopted different frameworks. The framework of CT adopted in the present work is based on that presented by [20] and by the Google Exploring Computational Thinking website and education resources [21] which conceptualizes CT as having several key components including Abstraction, Algorithm Design, Evaluation, Generalization, Pattern Recognition, and Problem Decomposition. These components are not intended to be seen as a final declaration of what does and does not constitute CT or an exhaustive list of all its components. Rather, they have been identified as generally agreed upon components that are representative of the larger set of concepts and skills that have been included in various definitions of CT.

The purpose of this study is to report the development and validation work that has been conducted on the Computational Thinking Concepts and Skills Test (CTCAST). The CTCAST is an 18-item test that is half multiple-choice items and half multiple-

select items (e.g., items with an instruction such as, "mark all that apply"). It was created because there are few CT assessments for undergraduates that have been psychometrically evaluated and made available to the CS education research community. Furthermore, a test of CT, as opposed to a test of CS knowledge and skills, is likely to be useful to researchers working with students outside the traditional CS curriculum in courses such as CS0, informatics, digital humanities, etc.

The remainder of this paper first reviews existing measures of CT in the published research literature and summarizes prior work on scoring schemes for multiple-select test items. Section 3 describes the process used in developing the CTCAST. Sections 4-6 present the results of item analyses and comparisons of three scoring schemes and discuss the results of the item analyses and comparisons. Finally, conclusions are presented in Section 7.

2 RELATED RESEARCH

2.1 Other Tests of Computational Thinking

Due to the lack of validated measures of CT, multiple groups have set out to develop and validate tests that can be used to assess computational thinking. Next, six existing measures are described.

2.1.1. Tests for K-12 Populations. The *Computational Thinking Test (CTt)* [12, 13] was developed for Spanish students from fifth to tenth grade. It is a 28-item multiple-choice test that aims to measure students' development level of CT. The test covers basic directions and sequences, loops (repeat times), loops (repeat until), simple conditionals (if), complex conditionals (if/else), while conditionals, and simple functions. It is directed at beginner-level CS students and assumes participants have no prior knowledge of computer programming. The test is administered on-line and takes approximately 45-minute to complete. Acceptably high reliability estimates (> .70) for the CTt have been reported, and initial validity evidence comes from the weak relationship between scores on the CTt and other tests of mental abilities such as verbal and spatial reasoning. However, CTt scores were found to correlate strongly with a test of general mental ability (r = .669), suggesting the test largely taps *general intelligence.* Additionally, the test was written in Spanish and it appears that to date, no English translation has been created or evaluated.

[5] also created a test for middle-school students. It was constructed as part of a study of a computing education module, and was partially adapted from multiple sources, including 22 items from a version of the Israeli national exam. The exam included questions related to definitions of key computational terms (algorithm, variable, initialization, conditional, Boolean variable and loop), understanding of algorithms using snippets of Scratch code (a beginner-level coding environment), and debugging snippets of Scratch code. This assessment does not cover just CT topics, as it also includes computer science skills and even environment-specific skills. Reliability and validity information for the test was not reported.

A final example of a testing CT at the K-12 level comes from the Bebras Challenge [2, 6]. This annual international contest involves K-12 students taking CT tests appropriate for their age level. Each year, different test items are used, and the tests are given in many languages, making the Bebras Challenge a group of tests rather than *a* test. Impressively, the tests have been taken by more than one million students, over more than ten years. However, neither the tests nor reliability and validity information about them are published.

2.1.2. Tests for Undergraduate-level Populations. The *Foundational CS1* (FCS1) Assessment instrument [15–17] measures student learning in undergraduate-level introductory computer science education courses. The test contains 26 items and uses a multiple-choice format where a right answer is scored a one and a wrong answer is scored a zero. The questions on the test are written in pseudocode developed by the author to achieve programming language independence. The test is not publicly available to prevent participant bias. A test parallel to the FCS1 test has been created [10], and is called the *Second CS1* (SCS1) Assessment. The purpose of the SCS1 Assessment is to make a version of the FCS1 widely-accessible while protecting the participant bias of the original FCS1 assessment. The SCS1 is also language-independent and covers the same topics as the FCS1. The only published reliability estimates for the SCS1 and the FCS1 are internal consistency estimates reported by [10] and are lower than is generally considered acceptable for tests of knowledge and skills (i.e., < .60). Additionally, item analysis results have only been reported for the SCS1 [10], and for the sample in that study, the items tended to be difficult and have low discrimination.

The test created by [4] was administered to introductory computer science students at a South African university. They intended to measure students' CT abilities and contrasted the test results with student performance in the introductory computer science course. The test is comprised of 20 multiple-choice or short-answers questions and 5 'warm-up' questions that are not considered in the scoring. The 20 questions belong to one of six computation thinking classifications: *Processes & Transformations, Models & Abstractions, Patterns & Algorithms, Tools & Resources, Inference & Logic,* and *Evaluations & Improvements.* The test was taken using pen and paper with allotted space for rough-work and answers. Students' were given 90 minutes to complete the test. Psychometric characteristics of the test have not been made available, so quality of the test in terms of reliability and validity is unknown.

[8] created a classroom assessment for first-semester undergraduate students taking a Java programming course. The test contains 26 multiple-choice questions and is used to assess students' understanding of three primary computing concepts: basic object-oriented programming (*classes, instances, events, and methods*), basic program control constructs (*sequence, selection, and iteration*), and basic sorting and searching algorithms on arrays. The questions on control constructs and sorting and search algorithms are presented as Java code, and it is expected that students taking the test are familiar with Java; however, it is expected that the specific code used in the test has not been seen by the test-taker prior to its administration. This test was closely aligned with a first-semester programming curriculum, and is unlikely to meet the needs of researchers looking for a test of CT.

2.1.3 Summary. The tests reviewed in this section all address computing knowledge and skills, but they vary in the degree to which they measure mastery of formal CS content versus deeper, more general principles of CT. Across the set of tests created for undergraduates, surprisingly little has been published about the psychometric quality of the tests, making it difficult to assess their quality and fitness for use in research or classroom assessment. The tests reported in [5, 12, 13] were created for K-12 populations and are not necessarily appropriate for assessing undergraduate students' CT.

2.2 Multiple-Response Items and Scoring Methods

Selected-response items are assessment items that present the respondent with a set potential answer options (e.g., multiple-choice items, true/false items), rather than requiring the respondent to generate an answer on their own (as is the case with short-answer or essay questions). Multiple-select and multiple-true/false items are similar versions of selected-response items that allow for or require multiple responses within the same item. The stems of these items often contain phrases such as "mark all that apply" or "select all that are true."

Despite having been introduced several decades ago, there remains no consensus on how to best score multiple-response items [7, 18], though numerous scoring methods have been proposed and tested [1, 3, 7, 18]. There are four common scoring options that have been tested by multiple researchers. First, *all-or-none* scoring (also called *cluster scoring*, *rigid scoring*, or *multiple response scoring*), dictates that a response is scored as correct only if the individual provides a completely correct response pattern for that item. For example, an item with two correct options followed by two incorrect options would have the correct response pattern of < X X O O >. A response has to match that pattern exactly in order to be scored as 1, and any other response pattern (e.g., < X X X O >, < X O O O >, etc.) is scored as 0.

Second, *some-or-none* scoring specifies a minimum number of options that must be correct for the examinee to receive a point, and if that threshold is not met, no points are earned. This scoring system is more generous than all-or-none because it does not require a perfect response, but it does not introduce partial points.

Partial-credit scoring systems, the third class of scoring systems, have varying levels of protection against guessing correct answers. These systems set some minimum level at which partial credit will be awarded so that partially correct answers under that threshold receive no points. For example, a partial credit system with a required minimum of 50% might award 1.0 point for providing a completely correct answer, 0.5 points for providing an answer that is 50-99% correct, and 0 points for providing an answer that is less than 50% correct. In the most lenient of the partial-credit scoring systems, simple partial-credit (SPC) scoring each component of a response is judged as correct or incorrect, and each correct component is scored according to its proportion of the total number of options for that item. For an item with 4 options, each component is worth 0.25 points. So, with the previous example correct response pattern of < X X O O >, a matching response is scored as 1, the response patterns < X X X O > and < X O O O > are scored as 0.75, and any of the other possible partially or completely incorrect response patterns would be scored 0.75, 0.50, 0.25, or 0.0. Importantly, though, SPC scoring still treats each response as part of a single item, and not as separate items worth a full point each. Scoring each option as a single, full item is not recommended because of the dependence between items that share part or all of the stem [1, 19].

Fourth, in addition to all-or-none, some-or-none, and partial-credit scoring methods, [19] used two different scoring systems that focused on the number of options that were correct if true or marked, for multiple-true/false and multiple-select items, respectively. One system was an *all-true/marked-correct* where a point was awarded if all of the true/marked options were correct, regardless of the responses given to false/unmarked options. The other was a partial-credit system where the score given was the number of true/marked options that were correct was divided by the number of true/marked options in the item. The reasoning behind these options was that there is some evidence that less-knowledgeable examinees are more likely to leave blank an option for which they do not know they answer, and are therefore more likely to get those items correct.

Comparisons of different scoring methods have shown that *some-or-none and partial-credit scoring yield higher item means (i.e., more correct responses) than all-or-none scoring*, and *more generous some-or-none and partial-credit scoring methods tend to have higher item means than less generous methods* [3, 7]. As has been pointed out [18], the differences in item means are especially relevant when the absolute value of a test score matters (i.e., criterion-referencing), as is usually the case when assigning grades, but it is less important when scores are only used for relative comparisons (i.e., norm-referencing), as is more often the case in research contexts.

In terms of *test reliability*, [18] did not find any differences among six different scoring methods. In contrast, [19] found small increases (change in alpha < .01) in reliability for some partial-credit and some-or-none methods, relative to all-or-none scoring, and [3] and [7] found slightly larger increases for two different partial credit scoring methods (changes in alpha up to .08).

Findings related to *item discrimination* (i.e., corrected item-total correlations) have been mixed. [3] found no clear difference in discrimination among three scoring methods. [19] found that for some items, partial-credit scoring provided better discrimination than did some-or-none and all-or-none scoring, but this result was not consistent across all tested items. [7] reported the average discrimination for items scored with partial-credit methods were higher than the average for all-or-non scoring.

Overall, prior research suggests that if the scoring method has any impact on a test's psychometric properties, *partial-credit scoring methods are probably slightly better than all-or-none scoring.*

3 DEVELOPMENT PROCESS

Our research team initially developed a test of core CS1 content that was used to as a standard measure of learning across multiple semesters sections of CS1. This multiple-choice CS knowledge test, the Nebraska Assessment of Computing Knowledge (NACK), contained a combination of conceptual and application questions

that were written by computer science faculty. The psychometric properties of the test were evaluated, and through an iterative revision and psychometric evaluation process, the initial pool of 26 items was reduced to a set of 13 items. Additional information about the NACK can be found in [9, 14].

After multiple successful administrations of the NACK, it was determined that a test of CT that was less directly connected to CS1 content would be better suited for studies involving students in courses other than introductory level CS. We then set out to create a new test that would more purely measure components CT rather than CS1 content. Test items, written by computer science faculty, targeted one of the six aforementioned components of CT. For each component, both knowledge and application items were written. Two different types of knowledge items were written: *definition (or concept recall)* items and *instantiation (or understanding)* items. To test application, items focused on *problem solving that involved practicing* a specific component of CT.

An initial 18-item version of the CTCAST was administered electronically in the spring and fall semesters of 2017 (information on the pilot study is given in Section 3.1). Following analysis of the pilot data, the test was revised and administered during the fall 2018 and spring 2019 semesters (information on the sample is given in Section 4.1). Again, item analysis was conducted, and additional analyses to establish initial validity evidence were performed (see Section 4).

3.1 Pilot Study

The pilot version of the CTCAST was administered in class via Survey Monkey toward the end of the spring and fall 2017 semesters. Data collection took place as part of a larger study examining students' motivation, self-regulated learning, and engagement in CS courses. During these two semesters, students were recruited from five 100-level courses, including two honors courses, three 300-level courses, including one honors course, and two 400-level courses. Participants represented all levels of academic standing (first-year = 146, sophomore = 101, junior = 59, senior = 38, other/graduate = 37). Students who reported their standing as other/graduate were excluded from all analyses, because the CTCAST is intended for use with undergraduate populations. The undergraduate sample (N = 344) included 284 men and 60 women, approximating the overrepresentation of men in CS courses at the institution. For race/ethnicity, 218 students self-reported as White, 23 self-reported as Hispanic/Latino/a, 15 self-reported as Black, 4 self-reported as Native American, 98 self-reported as Asian/Pacific Islander, 5 self-reported Other, and 5 indicated they preferred to not answer. (The sum of race/ethnicity self-report does not equal the total sample size because participants were able to select multiple options.) Additionally, 197 participants were majoring or minoring in CS, 74 were considering a major or minor in CS, and 70 were not considering a major or minor in CS.

Poorly functioning items were identified through Classical Test Theory (CTT)-based item analysis, which was conducted in SPSS v. 25. Statistics of primary interest in evaluating items were difficulty (item mean), discrimination (corrected item-total correlation), and coefficient alpha-if-item-deleted. As a result of the item analysis, 6 items were identified as needing revision or removal, resulting in minor wording revisions to 3 items and more substantial revisions to 3 items. These more substantial revisions involved content changes to the item stem or at least one response option. Additionally, 9 items were initially presented as multiple-choice items with secondary response options such as "A: I and II; B: II and III; C: I, II, and III; D: only III," also known as Type K items [1]. These 9 items were reformatted as multiple-select items, also called Type X items [1]. The multiple-select options had check-boxes in place of the radio buttons shown for multiple-choice items, and each multiple-select item stem gave the instruction, "Mark all that are true." The full revised version of the CTCAST is available at https://cse.unl.edu/agents/ic2think/software.php.

4 THE COMPUTATIONAL THINKING CONCEPTS AND SKILLS TEST (CTCAST)

4.1 Procedure and Participants

The revised CTCAST and the NACK were again administered via the Qualtrics platform. The change in platforms was a result of personnel changes within the research team and was not due to any aspect of either platform, the test, or the research being conducted. Aside from the described revisions, the overall presentation of the survey was the same across the two platforms. The NACK was administered so that scores on that test could be used as a source of validity evidence for the CTCAST.

Participants ($N = 169$) were recruited at the end of the fall 2018 and spring 2019 semesters from CS1 classes. Three students did not complete the NACK; participants were included in all analyses for which the relevant data were available. Due to external constraints, data collection was limited to introductory classes during these terms. Most participants were underclassmen (first-year = 123, sophomore = 33, junior = 10, senior = 3), and male (men = 137, women = 32). For race/ethnicity, 123 students self-reported as White, 15 as Hispanic/Latino/a, 8 as Black, 23 as Asian/Pacific Islander, 4 indicated "other", and 4 selected "prefer not to answer". (Participants were able to select all that applied to them, so the total does not equal 169.) Fifty-nine participants were majoring or minoring in CS, 31 were considering a major or minor in CS, and 79 were not considering a major or minor in CS.

4.2 Scoring and Analysis

For all multiple-choice items, the correct answer was scored as 1, and incorrect answers were scored as 0. For the multiple-select items, three scoring options were tested: *all-or-none scoring*, *SPC scoring*, and a *weighted-partial credit (WPC) scoring method* that, to our knowledge, has not been tested before. The decision to use WPC scoring was based on the evidence that options that have to be left unmarked (or are false) are more likely to be answered correctly by less-knowledgeable examinees [11, 19] and was intended to lessen the impact of those items in the totals scores.

As part of preliminary analysis and data screening, the difficulty of items that had to be marked to be correct was compared to correct-if-unmarked items. In line with the findings of [11, 19], correct-if-unmarked items were easier than correct-if-marked

items. Because this use of weights in scoring multiple-select items is novel and the viability of the approach has not been established, it was decided that the choice of weights would prioritize mathematical and conceptual simplicity. As a result, the same weight was applied to all correct-if-marked options within an item and a different weight was applied to all correct-if-unmarked options within an item. Weights were selected so that within an item, correct-if-marked options were worth twice as much as correct-if-unmarked options and the sum of all partial points for each item still had a maximum of 1, which can be expressed in the set of equations

$$w_m m + w_u u = 1 \quad (1)$$

$$w_m = 2w_u \quad (2)$$

where w_m is the weight for marked-if-correct items, m is the number of options in an item that are correct if marked, w_u is the weight for correct-if-unmarked items, and u is the number of options in an item that are correct if unmarked.

The tests were first evaluated through item analysis. One item analysis was conducted with the multiple-choice items and multiple-select items scored with all-or-none scoring. The second item analysis was conducted with the multiple-choice items and multiple-select items scored with SPC scoring. The third item analysis was conducted with the multiple-choice items and multiple-select items scored with WPC scoring. Statistics of greatest interest in the item analyses were *item means* (an index of *difficulty*) *corrected item-total correlations* (an index of *discrimination*), and *alpha-if-deleted* (an index of the item's *impact on a reliability estimate*). *Correlations* were conducted to compare scores from the different methods to each other and to scores from the NACK. All analyses were conducted in SPSS V. 25.

5 ITEM ANALYSIS RESULTS

Two sets of item analyses were conducted. The first included the multiple-choice items and the multiple-select items scored with the all-or-none procedure. The second included the multiple-choice items and the multiple-select items scored with the simple-partial-credit scoring procedure. Item-level scores (rather than option-level scores) were used in the item analyses because, all other things equal, more items in a test will have a higher alpha, so option-level scores would inflate alpha. Furthermore, it has been reported that treating each response option as an individual item introduces excessive levels of item-dependence [1, 19] and can lead to an overestimation of item discrimination and total test information [19].

For the all-or-none scoring method, the coefficient alpha was 0.693. For the SPC method, the coefficient alpha was 0.739. For the WPC method, alpha was also 0.739. Item-level statistics for all item analyses are given in Table 1. Consistent with prior research [3, 7], multiple-select items were easier when SPC scoring was used than when all-or-none scoring was used, with all item means increasing by at least 0.30. Item means for WPC scoring were either between those of the all-or-none and SPC scores or equal

Table 1. Item Analysis Statistics for Two Item Analyses

	Mean			Standard Deviation			Corrected item-total correlation			Alpha-if-deleted			Change-in-alpha-if-deleted		
Item	A-N	SPC	WPC	A-N	SPC	WPC	A-N	SPC	WPC	A-N	SPC	WPC	A-N	SPC	WPC
MC															
2	0.76	---	---	0.43	---	---	0.405	0.449	0.421	0.67	0.71	0.72	-0.03	-0.03	-0.02
3	0.41	---	---	0.49	---	---	0.242	0.269	0.293	0.68	0.74	0.73	-0.01	0.00	-0.01
5	0.81	---	---	0.39	---	---	0.370	0.411	0.419	0.67	0.72	0.72	-0.02	-0.02	-0.02
6	0.70	---	---	0.46	---	---	0.205	0.283	0.283	0.69	0.73	0.73	0.00	-0.01	-0.01
8	0.54	---	---	0.54	---	---	0.233	0.218	0.200	0.69	0.74	0.74	-0.01	0.00	0.00
11	0.54	---	---	0.50	---	---	0.231	0.298	0.257	0.69	0.73	0.74	-0.01	-0.01	0.00
14	0.92	---	---	0.28	---	---	0.383	0.416	0.418	0.68	0.72	0.72	-0.02	-0.02	-0.02
15	0.47	---	---	0.50	---	---	0.256	0.263	0.258	0.68	0.74	0.74	-0.01	0.00	0.00
17	0.77	---	---	0.42	---	---	0.296	0.297	0.273	0.68	0.73	0.73	-0.01	-0.01	-0.01
MS															
1	0.36	0.73	0.73	0.48	0.26	0.26	0.314	0.456	0.476	0.68	0.72	0.72	-0.02	-0.02	-0.02
4	0.47	0.77	0.77	0.50	0.26	0.26	0.364	0.473	0.505	0.67	0.72	0.72	-0.02	-0.02	-0.02
7	0.09	0.58	0.56	0.29	0.25	0.24	0.108	0.277	0.438	0.69	0.73	0.72	0.00	-0.01	-0.02
9	0.47	0.83	0.60	0.50	0.21	0.19	0.256	0.332	0.220	0.68	0.73	0.74	-0.01	-0.01	0.00
10	0.33	0.76	0.62	0.47	0.22	0.22	0.300	0.400	0.412	0.68	0.73	0.72	-0.02	-0.01	-0.01
12	0.17	0.64	0.39	0.37	0.25	0.23	0.278	0.259	0.228	0.68	0.73	0.73	-0.01	-0.01	0.00
13	0.53	0.85	0.70	0.50	0.19	0.21	0.350	0.463	0.455	0.67	0.72	0.72	-0.02	-0.01	-0.02
16	0.40	0.71	0.71	0.49	0.28	0.28	0.194	0.256	0.284	0.69	0.73	0.73	0.00	-0.01	-0.01
18	0.11	0.58	0.47	0.32	0.26	0.24	0.259	0.418	0.450	0.68	0.72	0.72	-0.01	-0.02	-0.02
								Alpha							
Test	8.85	12.37	11.45	3.21	2.77	2.77	.693	.739	.739						

Note. A-N = All-or-none. SPC = Simple Partial Credit. WPC = Weighted Partial Credit. MC = multiple choice. MS = Multiple Select. Item means and standard deviations for multiple-choice items are the same under all scoring systems because their scoring was not impacted by the multiple-select scoring methods; other statistics are impacted by scoring of other items.

to the means for SPC (as was the case for three items with four marked-if-correct options, the equivalent of "all the above" in a multiple-choice question).

As can be seen in Table 1, a few items had slightly higher corrected item-total correlations (i.e., were more discriminating) under the all-or-none scoring, but most had higher correlations under the partial-credit methods. All-or-none scoring produced the highest discrimination values (by at least 0.01) for one multiple-choice item and one multiple-select item. SPC scoring produced the highest discrimination values (by at least 0.01) for two multiple-choice and one multiple-select items. WPC scoring produced the highest discrimination values (by at least 0.01) for one multiple-choice and six multiple-select items. Under all scoring systems, discrimination values for multiple-select items tended to be higher than those of multiple-choice items.

The changes in alpha that would result from removing items from the test shows little difference across scoring methods, and the impact of removing any item would be small. As a reminder, an increase in alpha if the item is deleted indicates the reliability of the test would be improve by removing the item.

6 COMPARISON OF SCORES

Sub-test scores were calculated for the group of multiple-choice items and for the group of multiple-select items using each scoring method. Correlations between the multiple-choice sub-test and the different multiple-select item sub-tests were calculated in order to examine the impact of scoring method on the relationship between the two groups of items. The correlation between the multiple-choice sub-test and the all-or-none sub-test was r = .41. The correlations were r = .58 for both the multiple-choice—SPC relationship and the multiple-choice—WPC relationship. The higher correlation for the partial-credit scoring options indicates *participants' performance on the two groups of items was more consistent when partial-credit scoring was used.*

Total scores produced by all three scoring methods were highly correlated (correlations shown in Table 2), indicating the different methods did not have much impact on how participants scored relative to one another once the scores for the multiple-choice and multiple-select items were combined. Scores were also correlated with scores from the NACK. The moderate correlations between the CTCAST scores and the NACK scores indicate the two instruments test similar, but not identical, domains, supporting the argument that CT is different than knowledge of basic CS concepts, and the two should be measured separately.

Table 2. Correlations among Tests and Scoring Methods

	A-N	SPC	WPC	NACK
A-N	---			
SPC	.942	---		
WPC	.945	.999	---	
NACK	.585	.568	.572	---

Note. All correlations significant, p < .001. A-N = All-or-none. SPC = Simple Partial Credit. WPC = Weighted Partial Credit. NACK = Nebraska Assessment of Computing Knowledge.

7 DISCUSSION AND CONCLUSIONS

The findings of this study indicate *the CTCAST is a sufficiently reliable test to be used in CS education research.* The convergence of scores on the CTCAST and the NACK also provide some initial validity evidence for the test, but additional research on the validity of the test is needed. Although the CTCAST has up to this point only be administered to undergraduate students, the authors believe it might be useful with some other groups, such as high school students, and further research should be conducted to determine the suitability of the test for other populations. Additionally, the current version should be more thoroughly evaluated with a broader range of undergraduate students, especially students in intermediate and upper-level CS courses.

Consistent with prior research on various scoring methods for multiple-select items [1, 3, 7, 18], *no single scoring method tested here emerged as clearly preferable, but the partial-credit scoring methods lead to more reliable (i.e., internally consistent) scores.* The WPC scoring method was able to partially adjust for the large increase in item means from the SPC scoring method, a feature that can be useful when one's goal is to keep item difficulty levels closer to 0.5, where there is the greatest potential for variability in scores. Additional research is needed to determine the optimal formulae for determining weights for these types of items. Because the inclusion of weights complicates the scoring process and it has not yet been determined if the weights used in this study are optimal, *it is currently recommended that the multiple-select items in the CTCAST be scored using the SPC method.*

The multiple-select items showed a clear advantage in terms of producing highly discriminating items, and as a result had a greater impact on the reliability of test scores than did the multiple-choice items. Interestingly, *using different scoring methods for the multiple-select items had an appreciable impact on some of the statistics of some of the multiple-choice items.* Prior studies have either have not discussed how different scoring options impact item statistics for multiple-choice items [19], have treated the multiple-choice and multiple-select items as separate subtests and have compared scoring methods at the subtest level [1, 3, 7], or have not had any multiple-choice items in the analyzed tests [18]. Given the generally higher discrimination values of multiple-select items across scoring methods, future research should further explore how the various scoring methods for multiple-select items can potentially improve the item characteristics of other types of test items.

The findings of this study demonstrate that the CTCAST is a psychometrically sound test that is suitable for use in CS education research. The test covers six key aspects of CT that overlap with, but are not identical to, core CS content. At this time, it is recommended that scores be calculated using SPC scoring for the multiple-select items and traditional right/wrong scoring for the multiple-choice items. However, the findings also indicate WPC scoring has potential as a useful scoring method for multiple-select items and should be investigated further.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation (Grants No. 0829647, 1431874, and 1122956).

REFERENCES

[1] M.A. Albanese and D. L. Sabers. 1988. Multiple true-false items: A study of interitem correlations, scoring alternatives, and reliability estimation. *Journal of Educational Measurement.* 25, 2 (Jun. 1988), 111–123. DOI:https://doi.org/10.1111/j.1745-3984.1988.tb00296.x.

[2] A. L. S. O. Araujo et al. 2019. How many abilities cn we measure in computational thinking? A study on Bebras Challenge. *In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19),* February 27-March 2, 2019, Minneapolis, MN, USA. *ACM*, New York, NY, USA, 545-551. https://doi.org/10.1145/3287324.3287405

[3] D. Bauer et al. 2011. Pick-N multiple choice-exams: a comparison of scoring algorithms. *Advances in Health Sciences Education.* 16, 2 (May 2011), 211–221. DOI:https://doi.org/10.1007/s10459-010-9256-1.

[4] L. Gouws et al. 2013. First year student performance in a test for computational thinking. *Proceedings of the 2015 SAICSIT Annual Conference* (East London, South Africa, Oct. 2013), 271–277.

[5] S. Grover et al. 2014. Assessing computational learning in K-12. *Proceedings of the 19th Annual Conference on Innovation and Technology in Computer Science Education* (New York, NY, Jun. 2014), 57–62.

[6] C. Izu, et al. 2017. Exploring Bebras tasks content and performance: A multinational study. *Informatics in Education* 16, 1 (2017), 39-59. DOI: 10.15388/infedu.2017.03

[7] F.-M. Lahner et al. 2018. Multiple true–false items: a comparison of scoring algorithms. *Advances in Health Sciences Education.* 23, 3 (Aug. 2018), 455–463. DOI:https://doi.org/10.1007/s10459-017-9805-y.

[8] R. Lister. 2005. One small step toward a culture of peer review and multi-institutional sharing of educational resources: a multiple choice exam for first semester programming students. *Proceedings of the 7th Australian Computing Education Conference* (Newcastle, Australia, 2005), 155–164.

[9] K. G. Nelson et al. 2015. Motivational and self-regulated learning profiles of students taking a foundational engineering course. *Journal of Engineering Education, 104*(1), 74-100.

[10] M. C. Parker et al. 2016. Replication, validation, and use of a language independent CS1 knowledge assessment. *Proceedings of the 2016 ACM Conference on International Computing Education Research* (New York, NY, 2016), 93–101

[11] M. Pomplun and M. D. H. Omar,. 1997. Multiple-mark items: An alternative objective item format? *Educational and Psychological Measurement.* 57, 6 (1997), 949–962.

[12] M. Roman-Gonzalez. 2015. Computational thinking test: Design guidelines and content validation. *EDULEARN15 Proceedings* (Barcelona, Spain, Jul. 2015), 2436–2444.

[13] M. Roman-Gonzalez, et al. 2017. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior.* 72, (2017), 678–691.

[14] D. F. Shell and L.-K. Soh. 2013. Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses. *Journal of Science Education and Technology, 22*, (2013), 899-913.

[15] A. E. Tew. 2010. *Assessing Fundamental Introductory Computing Concept Knowledge in a Language Independent Manner.* Georgia Institute of Technology.

[16] A. E. Tew and M. Guzdial. 2010. Developing a validated assessment of fundamental CS1 concepts. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (New York, NY, 2010), 97–101.

[17] A. E. Tew and M. Guzdial. 2011. The FCS1: a language independent assessment of CS1 knowledge. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (New York, NY, 2011), 111–116.

[18] F.-J. Tsai and H. K. Suen. 1993. A brief report on a comparison of six scoring methods for multiple true-false items. *Educational and Psychological Measurement.* 53, (1993), 399–404.

[19] S. Verbic. 2012. Information value of multiple response questions. *Psihologija.* 45, 4 (2012), 467–485. DOI:https://doi.org/10.2298/PSI1204467V.

[20] J.M. Wing. 2006. Computational thinking. *Communications of the ACM.* 49, 3 (2006), 33–35.

[21] Google for Education: Computational Thinking. *Exploring Computational Thinking.*